Implementation:

Data structure:

1. ‘Question’ is represented by this class:

```
class Question{
        String questionID;
        String tag;
        String difficulty;
}
```

2. ‘Questions Pool’ is an arrayList of Question.

3. ‘Requirements’ for each question set is an arrayList of Strings ([“Tag1”, “Tag2”, “Tag3”, “Tag4”, “Tag5”, “Tag6”, “Easy”, “Easy”, “Medium”, “Medium”, “Hard”, “Hard”])

4. ‘Quiz’ is an arrayList of String representing “Question ID” ([“Q1”, “Q2”]).
5. ‘Quizzes’ is an arrayList of ‘Quiz’

Algorithm

I've followed a greedy approach in generating a single quiz. Each quiz is prepared with two iterations over the questions pool.

Iteration 1:

First we iterate through the questions pool and choose those questions which fulfil both requirements - tag and difficulty level. Once a question is chosen, question is removed from the question pool and two elements (tag and difficulty level) get removed from the requirements list.

If our requirements are met we skip the second iteration (explained below) and move on to generating the next quiz.

Iteration 2:

If the first iteration is unable to fulfil our quiz requirements, we do another iteration of the questions pool. In this iteration, we choose those questions which fulfil exactly one requirement - either tag or difficulty level.

If our requirements are met, we move on to generating the next quiz.

If our requirements are not met and questions count exceeds 10, we stop the process and return count of already generated quizzes.

At the start of preparation of each quiz, we count the number of questions remaining in the questions pool. If this count is less than 10x(number of sets generated so far), we stop the process. This is done because we need to reserve questions in the pool to fill the quizzes generated so far.

Assumptions:

1. Number of questions in the input file need to be 600
2. Each line format in the input file: `Q1|EASY|Tag2`

Test Cases considered :

1. Tag = Tag1 for all 600 questions (input1.txt)
   Output: 0.
2. Question pool is generated randomly (input2.txt)
   Output: 60.
3. Only 6 questions has tag from 1,2,3,4,5,6 respectively. Other questions are tagged tag1 (input3.txt)
   Output: 1.